

SISTEM PENGATURAN KECEPATAN MOTOR *BRUSHLESS DC* MENGGUNAKAN PENGENDALI *LINEAR QUADRATIC REGULATOR* (LQR)

Nor Rochmad Hadi Prasetya

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
email : norprasetya@mhs.unesa.ac.id

Muhamad Syariffuddin Zuhrie

Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
Email : zuhrie@unesa.ac.id

Abstrak

Teknologi yang berkembang pada saat ini dalam rangka memenuhi kebutuhan dan memudahkan kegiatan manusia semakin banyak. Salah satu hasil dari perkembangan teknologi adalah motor DC. Tujuan pada penelitian ini yaitu menerapkan pengendalian pada motor *brushless DC* agar dapat mengikuti *setpoint* seperti yang diinginkan. Dan pengendali yang digunakan adalah pengendali *Linear Quadratic Regulator* (LQR). Pengendali LQR memiliki kesederhanaan, dan kemudahan untuk mendapatkan suatu pengendalian karena hanya memerlukan nilai Q dan R yang akan menghasilkan parameter K (*gain feedback*) dan L (*setpoint tracking*). Perancangan *hardware* pada sistem ini adalah motor *brushless DC* dengan model A2212/10T sebagai *plant* dan *tachogenerator* sebagai sensor atau pembanding dengan *setpoint* agar keluaran dari motor *brushless DC* dapat diatur sesuai keinginan dan bagian pengendalian yang digunakan adalah NI Elvis II+ sebagai *hardware* untuk *signal conditioner*. Hasil penelitian menunjukkan nilai parameter LQR $Q=10$, $R=1$. Dengan nilai parameter LQR tersebut mampu menghasilkan respon sistem yang stabil pada *setpoint 2 Volt* dengan nilai t_d sebesar 0,481 detik, nilai $t_r(5\%-95\%)$ sebesar 2,043 detik, nilai $t_r(10\%-90\%)$ sebesar 1,525 detik, nilai $t_s(5\%)$ sebesar 2,082 detik, nilai $t_s(2\%)$ sebesar 2,776 detik, nilai $t_s(0,5\%)$ sebesar 3,47 detik, nilai E_{ss} tanpa beban sebesar 5,15% dan nilai E_{ss} dengan beban sebesar 8,45%.

Kata Kunci: Motor *Brushless DC*, LQR, LabVIEW, Pengendali Kecepatan

Abstract

Technology that is developing at this time in order to meet the needs and facilitate more human activities. One result of technological developments is DC motors. The purpose of this study is to apply control on a DC brushless motor so that it can follow the setpoint as desired. And the controller used is the Linear Quadratic Regulator (LQR) controller. The LQR controller has simplicity, and the ease of getting a control because it only requires the Q and R values which will produce the K (gain feedback) and L (setpoint tracking) parameters. Hardware design in this system is a DC brushless motor with A2212 / 10T model as a plant and tachogenerator as a sensor or comparator with a setpoint so that the output of the brushless motor DC can be adjusted as desired and the control part used is the Nvis Elvis II + as a signal conditioner hardware. The results showed that the LQR parameter value $Q=10$, $R=1$. With the LQR parameter value capable of producing a stable system response at 2 Volt setpoint with a t_d value of 0.481 seconds, the t_r value (5% -95%) was 2.043 seconds, the value t_r (10% -90%) of 1.525 seconds, t_s (5%) of 2.082 seconds, t_s (2%) of 2.776 seconds, t_s (0.5%) of 3.47 seconds, E_{ss} value of no load amounting to 5.15% and E_{ss} value with a load of 8.45%.

Keywords: Brushless DC Motor, LQR, LabVIEW, Speed Control

PENDAHULUAN

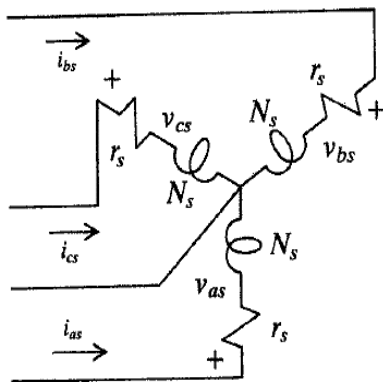
Untuk memenuhi kebutuhan dan memudahkan kegiatan manusia. Maka dikembangkan segala bentuk teknologi seperti halnya motor. Perkembangan motor dalam industri salah satunya adalah mesin *Computerized Numerical Control* (CNC). Diantara beberapa jenis motor, motor DC konvensional dikenal dengan karakteristiknya sangat bagus. Motor DC konvensional, memiliki beberapa kelemahan seperti perawatan rutin komutator, penggantian *brush* secara periodik dan biaya awal yang tinggi (W. R. Pearson, dkk, 1984). Maka, untuk mengurangi biaya

perawatan dari motor DC konvensional digunakanlah motor *brushless DC* yang memiliki efisiensi tinggi, kecepatan yang tinggi dan dapat divariasikan dengan biaya perawatan yang rendah (C. L. Xia, 2012). Dalam penelitian ini bertujuan untuk merancang pengendalian kecepatan motor *brushless DC* berbasis LabVIEW. Dimana LabVIEW digunakan sebagai program dan antarmuka dari grafik dan juga media pembuatan diagram block dan memunculkan hasil pengendalian kecepatan motor *brushless DC* menggunakan pengendali LQR.

KAJIAN PUSTAKA

Motor Brushless DC

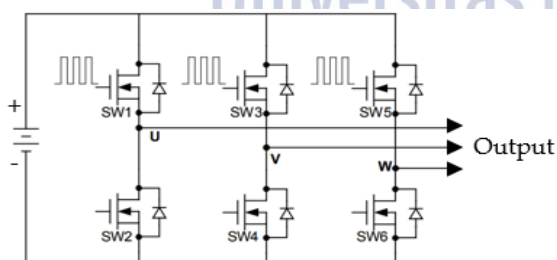
Motor *brushless DC* adalah motor sinkron dengan magnet permanen dan sebuah rangkaian elektronika (Inverter) untuk mengendalikan arus armature yang ada. Motor *brushless DC* kadangkala dimodelkan sebagai suatu motor DC 2 arah karena armature-nya ada di stator dan magnet berada pada rotor dan karakteristik operasinya mirip dengan motor DC. Daripada menggunakan suatu komutator mekanik seperti yang ada pada motor DC konvensional, motor *brushless DC* menggunakan komutasi elektronik yang menjadikannya motor yang minim perawatan (Azzumar, 2012). Gambar 1 menunjukkan rangkaian ekuivalen dari motor *brushless DC*.



Gambar 1. Rangkaian Ekuivalen Motor *Brushless DC* (Sumber : Krause, 2002)

Driver Motor *Brushless DC* (Inverter)

Cara kerja dari *inverter* untuk motor *brushless DC*, yaitu setiap mosfet aktif selama 180° terhadap siklus gelombang, hanya waktu pengaktifan mosfet masing-masing fasa dibuat berbeda 120°. Pergesaran fasa ini dihasilkan dalam urutan pengaktifan, mosfet diaktifkan pada suatu *interval* yang teratur dan terus menerus sebesar 60° secara berurutan, untuk menyelesaikan satu siklus dari gelombang tegangan keluaran (360°) (Azzumar, 2012). Gambar 2 menunjukkan skema umum *Driver Motor Brushless DC*.



Gambar 2. Skema Umum *Driver Motor Brushless DC* (Sumber : Zhao, dkk, 2011)

Tachogenerator

Tachogenerator adalah sensor yang sering digunakan untuk sensor kecepatan angular. *Tachogenerator*

merupakan sebuah generator kecil yang membangkitkan tegangan DC ataupun tegangan AC. Dari segi eksitasi tachogenerator dapat dibangkitkan dengan eksitasi dari luar atau imbas elektromagnet dari magnet permanen (Ridwan, 2012).

LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) adalah suatu *software* yang digunakan untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali dan instrumentasi, serta otomatisasi industri. Kelebihan LabVIEW dibandingkan dengan bahasa pemrograman lainnya adalah bahasanya yang mudah dipahami, pembuatan program mudah, waktu pembuatan program singkat, memiliki integrasi dengan ribuan *hardware* dan ratusan *library* yang siap digunakan, dapat menangani beberapa instruksi sekaligus, bersifat modular, telah terbukti andal, *powerfull*, dan fleksibel.

Analisis Karakteristik Orde Satu

Dari model matematis sebuah sistem, orde dari suatu sistem dapat dilihat dari besar pangkat variabel s (dalam transformasi Laplace). Suatu sistem dikatakan ber-orde satu jika fungsi alihnya mempunyai variabel s dengan pangkat tertinggi satu. Bentuk fisisnya bisa berupa rangkaian listrik *RC*, sistem termal, atau sistem lainnya (Ogata, 2010). Model sistem orde satu secara matematis dapat dituliskan sebagai berikut :

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1} \quad (1)$$

Keterangan :

$C(s)$ = Output sistem .

$R(s)$ = Input sistem.

K = Gain Overall.

$\tau s + 1$ = Waktu yang dibutuhkan untuk mencapai 63,2% (detik) pada orde satu.

Untuk menentukan parameter K (Gain Overall) jika sistem linier, maka hubungan Y_{ss} dengan X_{ss} dapat dituliskan sebagai berikut :

$$K = \frac{Y_{ss}}{X_{ss}} \quad (2)$$

Keterangan :

X_{ss} = Setpoint.

Y_{ss} = Hasil Respon.

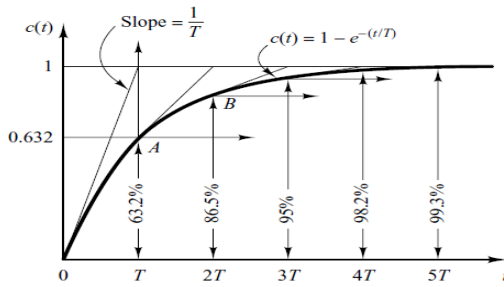
Untuk menentukan konstanta waktu (τ) dapat dihitung melalui respon keluaran sistem ketika mencapai 63,2% dari hasil akhirnya $C(\tau)$ dapat dihitung sebagai berikut :

$$C(\tau) = 0.632 \times Y_{ss} \quad (3)$$

Keterangan :

$C(\tau)$ = Keluaran Sistem ketika 63,2% dari hasil akhir.

Bentuk kurva tanggapan eksponensial orde satu ditunjukkan pada Gambar 3.



Gambar 3. Kurva Tanggapan Eksponensial Orde Satu (Sumber : Ogata, 2010)

Rumus analistik untuk perhitungan parameter respon dari orde pertama sebagai berikut :

1. Waktu tunda (td)

$$td = \tau \ln 2 \quad (4)$$

2. Waktu naik (tr)

- a. Waktu naik (5%-95%)

$$tr = \tau \ln 19 \quad (5)$$

- b. Waktu naik (10%-90%)

$$tr = \tau \ln 9 \quad (6)$$

3. Waktu Tunak (ts)

- a. Waktu tunak (0,5%)

$$ts = 5\tau \quad (7)$$

- b. Waktu tunak (2%)

$$ts = 4\tau \quad (8)$$

- c. Waktu tunak (5%)

$$ts = 3\tau \quad (9)$$

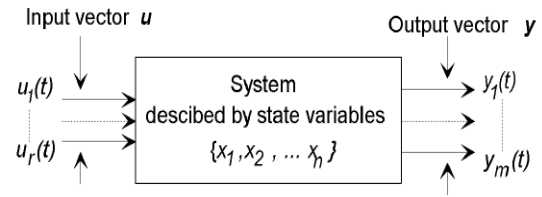
4. Persentase Error Steady State (Ess)

$$Ess = \left| \frac{y_{ss} - x_{ss}}{x_{ss}} \right| \times 100\% \quad (10)$$

State-Space

State-space merupakan metode analisis untuk sebuah sistem kendali yang kompleks. Metode ini digunakan untuk menganalisa sistem kendali dengan input banyak dan output banyak atau disebut MIMO (Multiple Inputs and Multiple Outputs). Pada model state-space keadaan internal (x) dari sistem digunakan untuk memprediksi output y(t). sehingga y tidak lagi bergantung hanya pada input sistem tetapi juga pada keadaan internal sistem.

Sistem input dan output pada state-space ditunjukkan pada Gambar 4.



Gambar 4. Sistem Input dan Output pada State-Space (Sumber : Ogata, 2010)

Persamaan model state-space dituliskan sebagai berikut (Ogata, 2010) :

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (11)$$

Keterangan :

A(t) = Matrix state

B(t) = Matrix input

C(t) = Matrix output

D(t) = Matrix direct-transmission

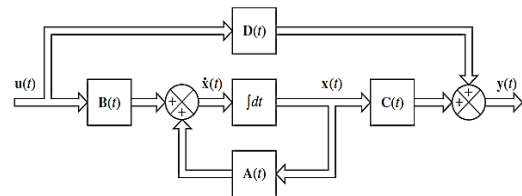
u(t) = Input system

x(t) = State system

y(t) = Output System

$\dot{x}(t)$ = Differential x

Diagram blok state-space ditunjukkan pada Gambar 5.



Gambar 5. Diagram Blok State-space (Sumber : Ogata, 2010)

Linear Quadratic Regulator (LQR)

Pada desain LQR dengan menggunakan persamaan matematis metode ini memiliki rumusan cost function sebagai berikut :

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (12)$$

Keterangan :

J = Cost function

Q = Faktor pembobotan state (matrix semidefinite positif)

R = Bobot faktor variabel kontrol (matrix definit positif)

[]^T = Transpose

. Untuk desain pengendali LQR, Langkah pertama adalah memilih matriks bobot nilai Q dan R. Masukkan R lebih berat daripada state sementara ketika nilai bobot Q state lebih dari input. Kemudian umpan balik K dapat dihitung dan respon loop sistem tertutup dapat ditemukan dengan simulasi.

Rumus pengendali LQR sebagai berikut :

$$u = -Kx \quad (13)$$

Keterangan :

K = Gain Feedback

Dari indeks persamaan *state space system* dan indeks performansi didapat nilai matriks K yang optimal untuk indeks performansi dengan rumus sebagai berikut :

$$K = R^{-1} B^T P \quad (14)$$

Keterangan :

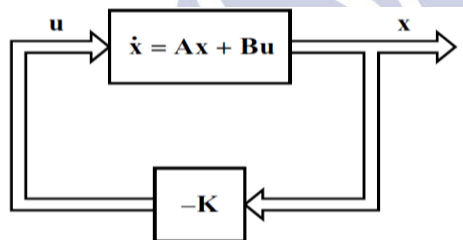
P = Variable Algebraic Riccati Equation (ARE)

$[\]^{-1}$ = Invers

Persamaan *Algebraic Riccati Equation* (ARE) untuk mendapatkan nilai P sebagai berikut:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \quad (15)$$

Dalam perancangan teknik kontrol optimal *Linear Quadratic Regulator* (LQR), setelah matriks P diketahui maka nilai P tersebut disubstitusikan kedalam Persamaan 14 sehingga didapatkan nilai matriks optimal K . Diagram blok pengendali LQR ditunjukkan pada Gambar 6.



Gambar 6. Diagram Blok Pengendali LQR (Sumber : Ogata, 2010)

Permasalahan yang ditemui dalam bidang kendali yaitu tidak hanya menyetabilkan *system*, tetapi bagaimana keluaran *system* mengikuti perubahan *setpoint* atau referensi yang ditentukan. Dalam hal ini jika diinginkan keluaran *plant* (x) sama dengan *input* referensi maka perlu dirancang sistemnya menggunakan *setpoint tracking* (Khijwania, 2012). Dari persamaan 13 tanpa pengendali *setpoint*, maka didapatkan bentuk umum rumus kendali *setpoint tracking* sebagai berikut:

$$u = -Kx + Lr \quad (16)$$

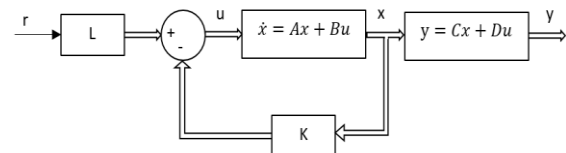
Keterangan :

L = Gain *setpoint tracking*

Dengan rumus L sebagai berikut :

$$L = (-C(A - BK)^{-1}B)^{-1} \quad (17)$$

Diagram blok pengendali LQR ditunjukkan pada Gambar 7.

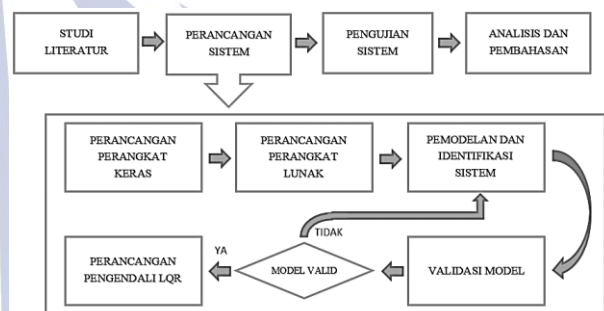


Gambar 7. Diagram Blok Pengendali LQR dengan *Setpoint Tracking* (Sumber : Khijwania, 2012)

METODE PENELITIAN

Rancangan Penelitian

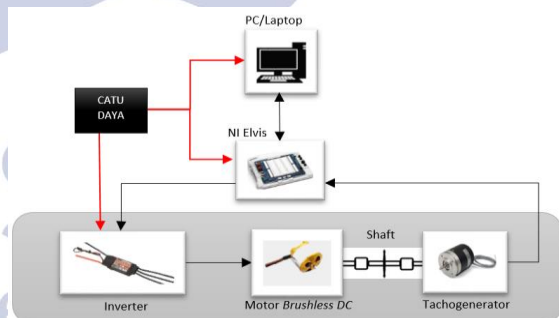
Tahapan perancangan penelitian ini secara garis besar dijelaskan dalam Gambar 8.



Gambar 8. Diagram Alir Rancangan Penelitian (Sumber : Data Primer, 2018)

Rancang Bangun Hardware

Desain system pengaturan kecepatan motor *brushless DC* ditunjukkan pada Gambar 9.



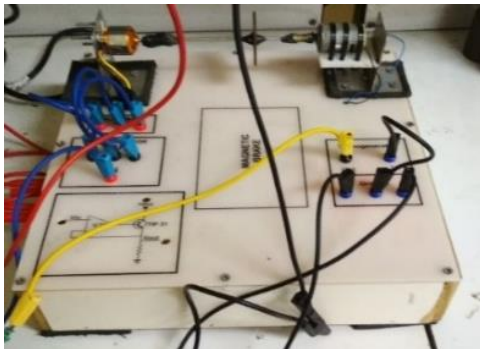
Gambar 9. Desain Sistem Pengaturan Kecepatan Motor *Brushless DC* (Sumber : Data Primer, 2018)

Motor *brushless DC* digunakan sebagai plan pada sistem yang digunakan dan dipasang *shaft* yang terhubung dengan *tachogenerator*. *Tachogenerator* digunakan sebagai pembaca kecepatan dari motor *brushless DC*. NI Elvis digunakan sebagai *Signal Conditioner*. *Inverter* digunakan sebagai *driver* untuk motor *brushless DC*. Inti otak pada sistem ini berada pada PC/Laptop dengan menggunakan *software* LabVIEW, dengan program yang digunakan adalah untuk me-monitoring kecepatan, mengatur *setpoint*, merubah *variabel* pengendali LQR, dan

Sistem Pengaturan Kecepatan Motor Brushless DC Menggunakan Pengendali Linear Quadratic Regulator (LQR)

switch untuk menentukan sistem menggunakan pengendali atau tidak.

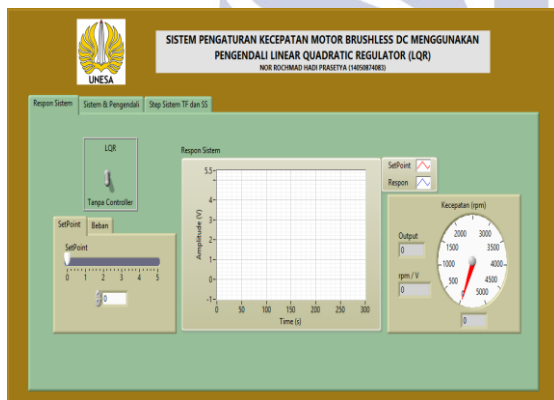
Bentuk fisik dari sistem pengaturan kecepatan motor *brushless DC* ditunjukkan pada Gambar 10.



Gambar 10. Bentuk Fisik Sistem Pengaturan Kecepatan Motor *Brushless DC* (Sumber : Data Primer, 2018)

Rancang Bangun Software

Software yang digunakan pada penelitian ini adalah LabVIEW 2014. LabVIEW 2014 berfungsi sebagai pusat pengendali dan juga berfungsi untuk mensimulasikan fungsi alih yang telah didapatkan sebelumnya. Tampilan *front panel* dapat dilihat pada Gambar 11.



Gambar 11. Tampilan *Front Panel* pada LabVIEW (Sumber : Data Primer, 2018)

HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas proses kalibrasi sensor *Tachogenerator*, pemodelan sistem, validasi model, perancangan kontroler LQR, dan pengujian respon sistem.

Kalibrasi Sensor *Tachogenerator*

Proses kalibrasi sensor *tachogenerator* dilakukan agar didapatkan nilai RPM *brushless DC* motor dalam setiap satuan volt yang dikeluarkan oleh sensor *Tachogenerator*. Alat bantu kalibrasi menggunakan *tachometer digital* tipe DT-2234C milik lab. kendali Universitas Negeri Surabaya.

Tabel kalibrasi ditunjukkan pada Tabel 1.

Tabel 1. Data Kalibrasi Tachogenerator (Sumber : Data Primer, 2018)

| No | Tachometer Digital (RPM) | Tachogenerator (Volt) |
|----|--------------------------|-----------------------|
| 1. | 275 | 0,5 |
| 2. | 557 | 1 |
| 3. | 850 | 1,5 |
| 4. | 1200 | 2 |
| 5. | 1375 | 2,5 |

Setelah mendapatkan data kalibrasi, dapat ditentukan nilai RPM setiap kenaikan tegangan pada *tachogenerator* yang telah dihitung yaitu rata-rata 557 RPM per 1V tegangan pada *tachogenerator*.

Pemodelan sistem

Pemodelan sistem dilakukan menggunakan metode pendekatan karakteristik sistem orde-1 dengan parameter yang dibutuhkan adalah K dan τ . Metode ini dilakukan bertujuan untuk mendapatkan model atau fungsi alih dari sistem. Fungsi alih tersebut digunakan sebagai acuan untuk merancang parameter dan simulasi untuk mengetahui respon sistem. Pada skripsi ini dilakukan pemodelan dengan 5 *setpoint* berbeda, yaitu 1,5V, 1,75V, 2V, 2,25V dan 2,5V. Menggunakan persamaan 1 dengan *setpoint* 2V didapatkan fungsi alih sebagai berikut :

$$G(s) = \frac{1,585}{0,658s+1}$$

Keterangan:

$G(s)$ = Fungsi alih pemodelan sistem

Validasi Sistem

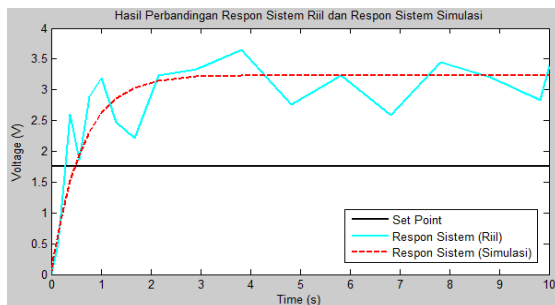
Validasi sistem dilakukan untuk mendapatkan model atau fungsi alih dari sistem yang memiliki respon dengan tingkat error. Validasi model juga berfungsi untuk menentukan apakah fungsi alih yang telah didapatkan benar dan dapat digunakan sebagai model dari *plant* riil. Fungsi alih tiap *setpoint* beserta error rata-rata ditunjukkan pada Tabel 2.

Tabel 2. Fungsi Alih tiap *Setpoint* (Sumber : Data Primer, 2018)

| Setpoint | Fungsi Alih | Error Rerata |
|----------|---------------------------------|--------------|
| 1,5 V | $G(s) = \frac{1,94}{0,651s+1}$ | 0,441 |
| 1,75 V | $G(s) = \frac{1,845}{0,601s+1}$ | 0,345 |
| 2 V | $G(s) = \frac{1,585}{0,658s+1}$ | 0,466 |
| 2,25 V | $G(s) = \frac{1,443}{0,713s+1}$ | 0,481 |
| 2.5 V | $G(s) = \frac{1,413}{0,601s+1}$ | 0,461 |

Dari data fungsi alih yang diperoleh dipilih fungsi alih dengan *error* rerata terkecil yaitu 0,345. Fungsi alih dengan *setpoint* 1,75 V dipilih sebagai acuan untuk mendesain parameter pada kondisi *open loop* riil dan simulasi.

Perbandingan respon dari fungsi alih dengan respon sistem riil dan simulasi ditunjukkan pada Gambar 12.



Gambar 12. Perbandingan Respon Sistem Riil dan Respon Sistem Simulasi
(Sumber : Dokumen Pribadi, 2018)

Sebelum merancang parameter dari pengendali LQR, dihitung parameter dari respon sistem riil *plant* secara *open loop* pada *setpoint* 1,75 V dengan nilai $\tau = 0,601$ detik. Parameter respon *plant* riil dengan *setpoint* 1,75V secara *open loop* ditunjukkan pada Tabel 3.

Tabel 3. Parameter Respon Riil dengan *Setpoint* 1,75V
(Sumber : Data Primer, 2018)

| Parameter Respon | Nilai |
|---------------------------|-------------|
| Waktu Tunda (td) | 0,417 detik |
| Waktu Naik (tr = 5%-95%) | 1,77 detik |
| Waktu Naik (tr = 10%-90%) | 1,321 detik |
| Waktu Tunak (ts = 5%) | 1,803 detik |
| Waktu Tunak (ts = 2%) | 2,404 detik |
| Waktu Tunak (ts = 0,5%) | 3,005 detik |
| Error SteadyState (Ess) | 81,086% |
| Maximum Overshoot (MO) | - |

Perancangan Pengendali LQR

Perancangan pengendali LQR dilakukan untuk mendapatkan parameter dari pengendali LQR yaitu K (*LQR*) dan L (*setpoint tracking (feed forward)*). Berikut adalah cara untuk mendapatkan nilai parameter untuk pengendali LQR :

Mengubah fungsi alih ke bentuk *state-space*

Untuk merubah bentuk fungsi alih ke bentuk *state-space* salah satu cara yang mudah untuk mengubah bentuk dari fungsi alih adalah dengan menggunakan *mathscript* yang ada pada program LabVIEW. Menggunakan persamaan 1 dengan *setpoint* 1,75 V didapatkan fungsi alih sebagai berikut :

$$G(s) = \frac{1,845}{0,601s+1} \quad (18)$$

Menggunakan persamaan 18, maka didapatkan nilai num dan den sebagai berikut :

$$\begin{aligned} \text{num} &= 1,845 \\ \text{den} &= [0,601 \ 1] \end{aligned}$$

Keterangan :

num = Pembilang dari fungsi alih

den = Penyebut dari fungsi alih

Dengan menuliskan *syntax* [A,B,C,D] = *tf_to_ss*(num,den), sehingga didapatkan parameter *matrix state-space* sebagai berikut :

$$\begin{aligned} A &= -1,6639 \\ B &= 1 \\ C &= 3,0699 \\ D &= 0 \end{aligned}$$

Perhitungan parameter untuk pengendali LQR

Berikut perhitungan untuk mencari nilai K dan L :

Nilai *matrix* Q dan R :

Nilai Q dan R

$$\begin{aligned} Q &= 0,1, 1, \text{ dan } 10 \\ R &= 1 \end{aligned}$$

Dengan menggunakan nilai *state-space* dan nilai *matrix* Q dan R. Selanjutnya perhitungan untuk mencari nilai P dengan menyelesaikan Persamaan 15. Sehingga didapatkan nilai P (dengan nilai Q=1) sebesar :

$$\begin{aligned} P_1 &= 0,2774 \\ P_2 &= -3,6052 \end{aligned}$$

Setelah itu menyubstitusikan nilai P ke Persamaan 14 (nilai P yang digunakan adalah 0,2774, karena syarat nilai P adalah P memiliki nilai *semi-definit positive*), sehingga didapatkan hasil sebagai berikut :

$$K = 0,2774$$

Selanjutnya mencari nilai L dengan menggunakan Persamaan 17, sebagai berikut :

$$L = 0,6324$$

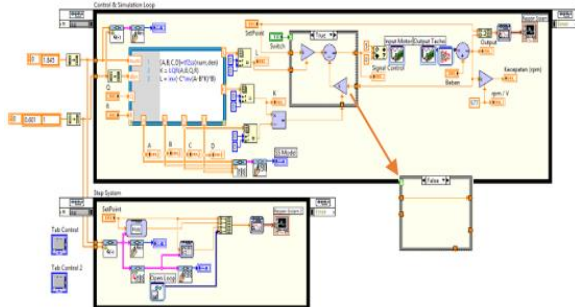
Nilai P, K dan L pada setiap perubahan nilai Q ditunjukkan pada Tabel 4.

Tabel 4. Nilai P, K dan L pada Setiap Perubahan Nilai Q
(Sumber : Data Primer, 2018)

| Q | R | P | K | L |
|-----|---|--------|--------|--------|
| 0,1 | 1 | 0,0298 | 0,0298 | 0,5517 |
| 1 | 1 | 0,2774 | 0,2774 | 0,6324 |
| 10 | 1 | 1,9094 | 1,9094 | 1,164 |

Implementasi Pengendali LQR

Setelah mendapatkan nilai K dan L, selanjutnya adalah mengimplementasikan pengendali LQR pada LabVIEW 2014. Diagram blok pada LabVIEW dengan pengendali LQR ditunjukkan pada Gambar 13.



Gambar 13. Diagram Blok pada LabVIEW dengan Pengendali LQR
(Sumber : LabVIEW, 2018)

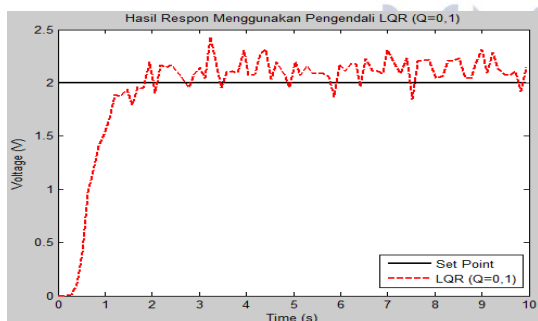
Untuk mengetahui perbedaan respon sistem sebelum dan sesudah ditambahkan pengendali LQR, maka dilakukan beberapa pengujian sebagai berikut :

Pengujian pengendali LQR dengan nilai $Q = 0,1$

Pengujian pengendali LQR dengan nilai $Q = 0,1$ dilakukan sebanyak tiga kali percobaan yaitu pengujian dengan *setpoint* tetap dan pengujian dengan beban. Setelah mendapatkan hasil dari masing-masing percobaan dapat dilanjutkan dengan menghitung parameter respon. Berikut adalah hasil pengujian dengan nilai $Q = 0,1$:

1. Pengujian dengan *Setpoint* Tetap

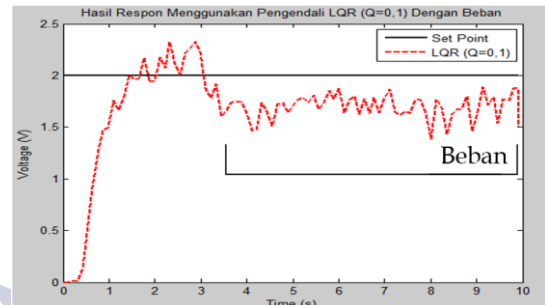
Dilakukan pengujian dengan *setpoint* tetap yaitu dengan *setpoint* 2V selama 10 detik. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=0,1$ ditunjukkan pada Gambar 14.



Gambar 14. Hasil Respon Menggunakan Pengendali LQR dengan Nilai $Q=0,1$
(Sumber : Dokumen Pribadi, 2018)

2. Pengujian dengan Beban

Dilakukan pengujian menggunakan beban dengan *setpoint* 2V dan beban dimulai dari detik ke-3 hingga detik ke-10. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=0,1$ dengan beban ditunjukkan pada Gambar 15.



Gambar 15. Hasil Respon Menggunakan Pengendali LQR ($Q=0,1$) dengan Beban
(Sumber : Dokumen Pribadi, 2018)

Ess (*Error SteadyState*) dengan beban menggunakan Persamaan 10 diperoleh nilai sebesar 15,2%.

3. Analisis Respon Sistem

Setelah ditambahkan pengendali LQR ($Q=0,1$) didapatkan nilai $\tau = 0,843$ detik. Parameter respon *plant* dengan *setpoint* 2V menggunakan pengendali LQR (tanpa beban) ditunjukkan pada Tabel 5.

Tabel 5. Parameter Respon Menggunakan Pengendali LQR ($Q=0,1$)
(Sumber : Data Primer, 2018)

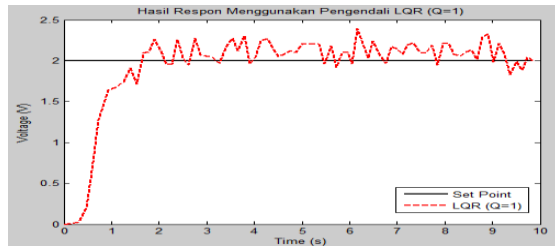
| Parameter Respon | Nilai |
|----------------------------------|-------------|
| Waktu Tunda (t_d) | 0,584 detik |
| Waktu Naik ($t_r = 5\%-95\%$) | 2,482 detik |
| Waktu Naik ($t_r = 10\%-90\%$) | 1,852 detik |
| Waktu Tunak ($t_s = 5\%$) | 2,529 detik |
| Waktu Tunak ($t_s = 2\%$) | 3,372 detik |
| Waktu Tunak ($t_s = 0,5\%$) | 4,215 detik |
| Error SteadyState (Ess) | 6,5% |
| Maximum Overshoot (MO) | - |

Pengujian pengendali LQR dengan nilai $Q = 1$

Pengujian pengendali LQR dengan nilai $Q = 1$ dilakukan sebanyak tiga kali percobaan yaitu pengujian dengan *setpoint* tetap dan pengujian dengan beban. Setelah mendapatkan hasil dari masing-masing percobaan dapat dilanjutkan dengan menghitung parameter respon. Berikut adalah hasil pengujian dengan nilai $Q = 1$.

1. Pengujian dengan *Setpoint* Tetap

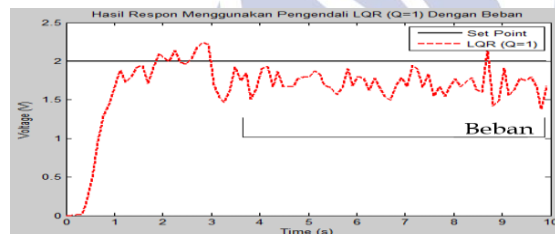
Dilakukan pengujian dengan *setpoint* tetap yaitu dengan *setpoint* 2V selama 10 detik. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=1$ ditunjukkan pada Gambar 16.



Gambar 16. Hasil Respon Menggunakan Pengendali LQR dengan Nilai $Q=1$
(Sumber : Dokumen Pribadi, 2018)

2. Pengujian dengan Beban

Dilakukan pengujian menggunakan beban dengan *setpoint* 2V dan beban dimulai dari detik ke-3 hingga detik ke-10. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=1$ dengan beban ditunjukkan pada Gambar 17.



Gambar 17. Hasil Respon Menggunakan Pengendali LQR ($Q=1$) dengan Beban
(Sumber : Dokumen Pribadi, 2018)

Ess (*Error SteadyState*) dengan beban menggunakan Persamaan 10 diperoleh nilai sebesar 14,2%.

3. Analisis Respon Sistem

Setelah ditambahkan pengendali LQR ($Q=1$) didapatkan nilai $\tau = 0,757$ detik. Parameter respon *plant* dengan *setpoint* 2V menggunakan pengendali LQR (tanpa beban) ditunjukkan pada Tabel 6.

Tabel 6. Parameter Respon Menggunakan Pengendali LQR ($Q=1$)
(Sumber : Data Primer, 2018)

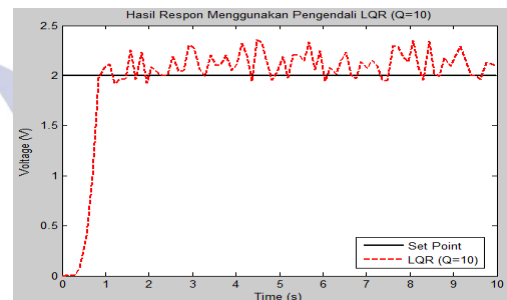
| Parameter Respon | Nilai |
|----------------------------------|-------------|
| Waktu Tunda (t_d) | 0,525 detik |
| Waktu Naik ($t_r = 5\%-95\%$) | 2,229 detik |
| Waktu Naik ($t_r = 10\%-90\%$) | 1,663 detik |
| Waktu Tunak ($t_s = 5\%$) | 2,271 detik |
| Waktu Tunak ($t_s = 2\%$) | 3,028 detik |
| Waktu Tunak ($t_s = 0,5\%$) | 3,785 detik |
| Error SteadyState (Ess) | 5,8% |
| Maximum Overshoot (MO) | - |

Pengujian pengendali LQR dengan nilai $Q = 10$

Pengujian pengendali LQR dengan nilai $Q = 10$ dilakukan sebanyak tiga kali percobaan yaitu pengujian dengan *setpoint* tetap dan pengujian dengan beban. Setelah mendapatkan hasil dari masing-masing percobaan dapat dilanjutkan dengan menghitung parameter respon. Berikut adalah hasil pengujian dengan nilai $Q = 10$.

1. Pengujian dengan *Setpoint* Tetap

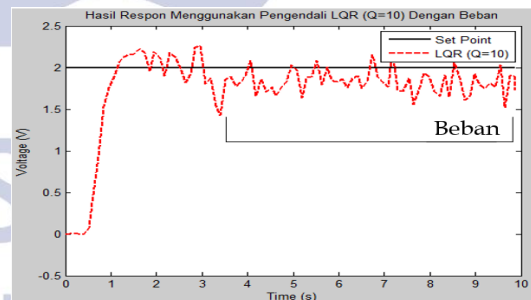
Dilakukan pengujian dengan *setpoint* tetap yaitu dengan *setpoint* 2V selama 10 detik. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=10$ ditunjukkan pada Gambar 18.



Gambar 18. Hasil Respon Menggunakan Pengendali LQR dengan Nilai $Q=10$
(Sumber : Dokumen Pribadi, 2018)

2. Pengujian dengan Beban

Dilakukan pengujian menggunakan beban dengan *setpoint* 2V dan beban dimulai dari detik ke-3 hingga detik ke-10. Hasil Respon menggunakan pengendali LQR dengan nilai $Q=10$ dengan beban ditunjukkan pada Gambar 19.



Gambar 19. Hasil Respon Menggunakan Pengendali LQR ($Q=10$) dengan Beban
(Sumber : Dokumen Pribadi, 2018)

Ess (*Error SteadyState*) dengan beban menggunakan Persamaan 10 diperoleh nilai sebesar 8,45%.

3. Analisis Respon Sistem

Setelah ditambahkan pengendali LQR ($Q=10$) didapatkan nilai $\tau = 0,694$ detik. Parameter respon *plant* dengan *setpoint* 2V menggunakan pengendali LQR (tanpa beban) ditunjukkan pada Tabel 7.

Tabel 7. Parameter Respon Menggunakan Pengendali LQR ($Q=10$)

(Sumber : Data Primer, 2018)

| Parameter Respon | Nilai |
|---------------------------|-------------|
| Waktu Tunda (td) | 0,481 detik |
| Waktu Naik (tr = 5%-95%) | 2,043 detik |
| Waktu Naik (tr = 10%-90%) | 1,525 detik |
| Waktu Tunak (ts = 5%) | 2,082 detik |
| Waktu Tunak (ts = 2%) | 2,776 detik |
| Waktu Tunak (ts = 0,5%) | 3,47 detik |
| Error SteadyState (Ess) | 5,15% |
| Maximum Overshoot (MO) | - |

PENUTUP

Simpulan

Berdasarkan penelitian yang dilakukan, maka didapatkan simpulan bahwa pengujian yang dilakukan menggunakan *software* Labview mampu mengolah data respon sistem riil dengan real time. Nilai $Q=10$ memiliki hasil yang lebih baik dari seluruh pengendali yang telah dilakukan. Masing-masing nilai dari parameter respon saat pengendalian LQR dengan nilai $Q=10$ yang terdapat pada Tabel 7 dengan nilai td sebesar 0,481 detik, nilai tr(5%-95%) sebesar 2,043 detik, nilai tr(10%-90%) sebesar 1,525 detik, nilai ts(5%) sebesar 2,082 detik, nilai ts(2%) sebesar 2,776 detik, nilai ts(0,5%) sebesar 3,47 detik, nilai Ess tanpa beban sebesar 5,15% dan saat diberikan gaya *magnetomotive* sebesar 500AT nilai Ess yang didapatkan sebesar 8,45%.

Saran

Berdasarkan simpulan diatas ada beberapa saran yang dapat dilakukan untuk pengembangan sistem agar hasil lebih maksimal yaitu model atau fungsi alih yang didapatkan masih kurang baik, sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini, menggunakan sensor *tachogenerator* yang lebih akurat dan presisi agar respon kecepatan yang didapat lebih baik dan sistem ini masih dapat dikembangkan dengan menggunakan metode pengendali yang lain seperti *fuzzy logic*, *MPC*, dan *PID* untuk mendapatkan hasil respon kecepatan yang lebih baik.

DAFTAR PUSTAKA

- Azzumar, Muhammad. 2012. *Pemodelan dan Desain Kendali Sistem Aktuator Kendali Sirip Berbasis Brushless DC Motor*. Depok: UI.
- Khijwania, Sunil. 2012. *Setpoint Tracking*. NPTEL, INDIA: IIT Guwahati.
- Krause, Paul. 2002. *Analysis of Electric Machinery and Drive Systems*. Second Edition. Purdue University. ISBN 0-471-14326-X
- Ogata, Katsuhiko. 2010. *Modern Control Engineering fifth edition*. New York: Prentice-Hall, Inc.
- Pearson, W. R. and Sen, Pares C. 1984. *Brushless DC Motor Propulsion Using Synchronous Motor for Transient System*. Kingston, Canada: IEEE Transection on Industrial Electronics.
- Pratama, Fauzi Yuda. 2018. *Rancang Bangun Pengendalian Kecepatan Brushless Motor DC Tipe A2212/10T 1400 KV Menggunakan Kontroler PID Berbasis Labview*. Surabaya: Unesa.
- Ridwan. 2012. *Sensor dan Tranduser*. Universitas Negeri Padang.
- Sibuea. Jordan Theja. 2018. *Implementasi Pengendali Kecepatan Brushless Motor DC Menggunakan Logika Fuzzy Berbasis GUI Labview 2014*. Surabaya: Unesa.
- Xia, C. L. 2012. *Permanent Magnet Brushless DC Motor Drives and Control*. China: Tianjin University.
- Zhao, Jian, dan Yu, Yangwei. 2011. *Brushless DC Motor Fundamentals*. MPS: The Future of Analog IC Technology.